

Центр Управления, руководство пользователя

Центр Управления, руководство пользователя

Архитектура Центра Управления

Установка Центра Управления содержит центральный экземпляр Центра Управления, отвечающий за веб-интерфейс, управление конфигурациями узлов и создание файлов конфигурации хостов и т. д. Исполнительным компонентом Центра Управления является Агент, который реализует функции управления TFTP, DHCP, DNS, Puppet, Puppet CA, Ansible и Salt. Агент устанавливается вместе с Центром Управления и помогает Центру Управления оркестрировать процесс ввода в эксплуатацию нового хоста. Типовая инсталляция развёртывает один Агент с полным набором функций управления непосредственно на хосте Центра Управления.

Центр Управления выполняет ряд шагов оркестрации при выполнении автоматизированного развёртывания хоста, которые варьируются в зависимости от выбранных вариантов интеграции, например, использование источников вычислительных ресурсов, инструмента управления конфигурацией и метода развёртывания (источник вычислительных ресурсов или PXE).

Пример: развёртывание через PXE с конфигурацией Salt.

В примере используются: вычислительный ресурс (oVirt), также применимо к Libvirt и VMware. Настройка сети (PXE) с оркестрацией DHCP и TFTP. Оркестрация DNS. Управление конфигурацией Salt. Также применимо к Puppet.

Шаги:

1. На странице создания нового хоста отображается конфигурация виртуальной машины по умолчанию, и могут быть применены профили вычислительных ресурсов.
2. Запрашивается неиспользуемый IP-адрес у Агента DHCP, связанного с подсетью.
3. Поле IP-адреса заполняется на странице создания нового хоста.
4. Страница создания нового хоста сохраняется.
5. Центр Управления связывается с вычислительным ресурсом для создания виртуальной машины.
6. Вычислительный ресурс создает виртуальную машину на гипервизоре.
7. MAC-адрес VM возвращается от вычислительного ресурса и сохраняется на хосте.
8. Создается резервирование на Агенте DHCP, связанном с подсетью.
9. Настраиваются DNS-записи:
 - Создается прямая DNS-запись Агентом DHCP, связанным с доменом.
 - Создается обратная DNS-запись Агентом DHCP, связанным с доменом.
10. Создается меню PXELinux для хоста на Агентом TFTP, связанным с подсетью.

11. Центр Управления связывается с вычислительным ресурсом для включения VM.
12. Вычислительный ресурс включает виртуальную машину.
13. Хост запрашивает аренду DHCP от DHCP-сервера.
14. Ответ на аренду DHCP возвращается с установленными параметрами TFTP (next-server, filename).
15. Хост запрашивает загрузчик и меню PXELinux с TFTP-сервера.
16. Возвращается меню PXELinux и установщик ОС для хоста через TFTP.
17. Установщик запрашивает шаблон/скрипт "provision" из Центра Управления.
18. Центр Управления обрабатывает шаблон и возвращает конфигурацию kickstart на хост.
19. Настройка автоподписи для Salt (или Puppet) добавляется на Агента Salt или Puppet CA.
20. Установщик уведомляет Центр Управления о успешном создании в постустановочном скрипте.
21. Меню PXELinux возвращается к шаблону "загрузка с локального диска".
22. Хост запрашивает свою конфигурацию у Salt или Puppet.
23. Хост получает соответствующую конфигурацию с использованием параметров и данных, определенных в Центре Управления.
24. Отчеты о конфигурации и факты отправляются из Salt или Puppet в Центр Управления и сохраняются.

Центр Управления может хранить информацию о сетевой настройке хоста, который он разворачивает, что позволяет настроить виртуальную машину, назначить правильные IP-адреса и сетевые ресурсы, а также правильно настроить операционную систему в процессе развертывания.

В этом разделе описаны доступные параметры для сетевых интерфейсов в форме создания нового хоста и их описание.

Опции интерфейса хоста:

1. Основной (Primary): каждый хост должен иметь один основной интерфейс, который используется в качестве имени хоста. Это считается основным именем хоста и обычно также используется для установки маршрута по умолчанию.
2. Установка (Provision): интерфейс, по которому будет выполняться установка операционной системы. Он используется для PXE (если применимо) с TFTP-меню, выполнения скриптов SSH и т. д. Этот интерфейс может отличаться от основного интерфейса и должен иметь доступ к Центру Управления и другим системам развертывания.
3. Управляемый (Managed): указывает, будет ли Центр Управления управлять созданием записей DNS, резервацией DHCP и настройкой интерфейса во время развертывания. Неуправляемые интерфейсы могут использоваться только в информационных целях.

Простой хост с одним интерфейсом, находящимся в одной сети, должен иметь установленное имя DNS, соответствующее имени хоста, а также флаги "Managed", "Primary" и "Provision" должны быть активированы. Это создаст один интерфейс с записями DNS и DHCP, через который будет настраиваться операционная система.

Хост с двумя или несколькими интерфейсами может иметь один интерфейс с активированным флагом "Primary" (например, "host.example.com") и другую сеть с активированным флагом "Provision" (например, "host-build.example.com"). Если оба интерфейса также управляемые, Центр Управления создаст записи DNS и DHCP для обоих интерфейсов. Однако для интерфейса "Provision" также будут установлены параметры next-server/filename для PXE. Также будет создано TFTP-меню (PXElinux) для MAC-адреса интерфейса "Provision", чтобы хост мог загрузиться по сети на этом физическом интерфейсе, а имя хоста будет назначено на основе интерфейса "Primary".

Когда Центр Управления развертывает хост на вычислительном ресурсе, он создает новый интерфейс на виртуальной машине для каждого интерфейса, указанного при создании хоста.

После создания Центр Управления считывает информацию о сети и сопоставляет созданные интерфейсы с указанным списком интерфейсов для хоста, сохраняя присвоенные MAC- и IP-адреса (в зависимости от типа вычислительного ресурса) в своей базе данных. Затем он продолжает оркестрацию, создавая записи DNS и DHCP и т. д. для прочих адресов, полученных с новой виртуальной машины. После завершения оркестрации он включает виртуальную машину.

Этап автоматического обнаружения интерфейсов позволяет избежать необходимости указывать MAC-адреса в случае, если хост создаётся на вычислительных ресурсах и MAC адрес не может быть известен заранее.

Параметры подсети

Подсети определяются в Центре Управления в разделе "Инфраструктура" > "Подсети" и имеют несколько опций, которые влияют на процесс развертывания хостов.

IPAM:

- DHCP будет обращаться к Агенту DHCP, проверяя назначенные адреса и резервации и предлагая свободный IP из диапазона.
- Внутренняя БД будет использовать список уже назначенных IP-адресов Центра Управления и не будет полагаться на DHCP Агент.
- Опция None отключает автоматическое предложение IP-адресов.

Режим загрузки: во время развертывания операционной системы шаблон будет настраивать интерфейс с использованием статического IP-адреса или DHCP в зависимости от значения этой настройки. Существуют различные комбинации параметров IPAM и режима загрузки, но наиболее распространены DHCP (IPAM) с DHCP (режим загрузки) и Internal DB с Static.

По умолчанию Центр Управления предоставляет фрагмент кода ("kickstart_networking_setup") для ОС на основе kickstart, который настраивает все управляемые сетевые интерфейсы после завершения основной установки ОС. Это можно использовать в разделе %post раздела kickstart.

Кортеж со сведениями о интерфейсах также доступен для Puppet через глобальный параметр ENC под названием "foreman_interfaces". Его можно использовать для дальнейшей настройки

сетевых служб хоста во время выполнения Puppet.

Управление Puppet

В этом разделе мы рассмотрим различные способы управления и взаимодействия с Puppet.

Окружения

Окружения Puppet отображаются в Центре Управления. Окружения Puppet обычно используются для разделения классов для разных типов хостов, что позволяет вносить изменения в модуль и тестировать его в одном окружении (например, development), прежде чем применять его в другом (например, production).

Создание окружений

Существует несколько способов создания окружений Puppet в Центре Управления.

Импорт из Puppet.

Центр Управления может обнаруживать все окружения и классы, настроенные на сервере Puppet, и импортировать их автоматически. В типовой инсталляции определено одно окружение Puppet - "production" - и уже наполнено стартовым набором классов, установленным с установочного носителя Центра Управления. В процессе эксплуатации вы наверняка захотите добавить свои классы Puppet или создать свои окружения. После внесения изменений на сервере Puppet из командной строки перейдите в раздел Настройки > Окружения Puppet и нажмите "Импорт окружений из ...". Центр Управления просканирует сервер Puppet через Агент и отобразит подтверждение обнаруженных изменений. Выберите изменения, которые вы хотите применить, и подтвердите.

Обратите внимание, что Агент обнаруживает только окружения, содержащие один или несколько классов Puppet, поэтому убедитесь, что на сервере Puppet развернут хотя бы один модуль Puppet, содержащий класс.

Создание вручную.

Чтобы создать окружение вручную, перейдите в раздел Настройки > Окружения Puppet и нажмите "Создать окружение Puppet". Дайте новому окружению имя и сохраните. Обратите внимание, что если окружение не существует на сервере Puppet, а затем вы выполните импорт (описан выше), Центр Управления попросит удалить это окружение. Наполнить окружение необходимо в интерфейсе командной строки стандартными методами Puppet.

Назначение окружений для хостов Это делается на странице редактирования хоста, на вкладке Host. Выбор окружения фильтрует видимые классы на вкладке Puppet Classes только по классам, принадлежащим выбранному окружению. Вы также можете массово назначить окружение для группы хостов: отметьте флажки нужных хостов в списке хостов, а затем выберите Change Environment в раскрывающемся меню Select Action в верхней части страницы.

Окружения с группами хостов.

Вы также можете назначить окружение группе хостов. Это действует как форма значения по умолчанию - при создании нового хоста и добавлении его в такую группу хостов окружение группы будет автоматически предложено и введено в соответствующее поле ввода.

Пользователь может изменить или удалить предложенное окружение и не применять его к новому хосту, но в типичной ситуации это экономит несколько кликов.

Классы

Классы Puppet импортируются с Puppet-сервера через кнопку Import на странице Puppet Classes. Они также могут быть созданы вручную и вручную связаны с набором окружений (для целей фильтрации).

Импорт классов.

Перейдите в раздел Настройки > Puppet Classes и нажмите кнопку Import. Будут импортированы только классы из модулей, а файлы манифестов Puppet должны быть валидными, чтобы их можно было разобрать с помощью Агента Puppet. Используйте команду puppet parser validate для проверки синтаксиса файлов манифестов Puppet когда добавляете свои модули чтобы убедиться, что модуль может быть импортирован в Центр Управления.

Столбец "Hosts" На странице Configure > Classes также есть столбец "Hosts". Этот столбец показывает количество хостов, к которым присвоен данный модуль/класс. Нажатие на это число покажет список хостов.

Игнорирование классов.

Часто модуль имеет такую структуру:

```
$ tree git/
git/
├── manifests
│   ├── init.pp
│   ├── install.pp
│   ├── params.pp
│   └── repo.pp
```

В этой ситуации Центр Управления предложит создать:

```
git
git::install
git::params
git::repo
```

Однако, если мы знаем, что подклассы не предназначены для прямого использования, а являются только частью внутренней структуры модуля, то мы захотим исключить их из механизма импорта, чтобы Центр Управления предложил импортировать только (в данном примере) git. Мы можем сделать это с помощью файла

```
/usr/share/foreman/config/ignored_environments.yml
```

Этот файл считывается при каждом импорте, и Центр Управления игнорирует изменения, которые соответствуют выражениям, перечисленным в файле. Впрочем, он не будет удалять уже существующие окружения или классы если правки в файл были внесены позднее.

Возможно игнорировать окружения целиком с помощью следующей конфигурации:

```
:ignored:  
- development  
- testenv
```

Классы могут быть игнорированы с использованием набора регулярных выражений - любой класс, соответствующий одному из них, не будет импортирован. Например, для приведенного выше примера мы можем настроить следующее:

```
:filters:  
- !ruby/regexp '/install$/'  
- !ruby/regexp '/params$/'  
- !ruby/regexp '/repo$/'
```

Для более сложной фильтрации можно использовать функции регулярных выражений, например, для игнорирования всех классов, кроме тех, которые начинаются с "role::", можно использовать следующий синтаксис:

```
:filters:  
- !ruby/regexp '/^(?!role::)/'
```

Назначение классов хостам.

Чтобы Puppet применял ваши классы, вам нужно назначить их хостам. Это можно сделать несколькими способами - лучший метод может варьироваться в зависимости от количества классов, которые вы планируете назначить, и от того, нужно ли изменять какие-либо параметры.

Назначение на отдельные хосты.

При редактировании хоста классы Puppet могут быть назначены непосредственно на вкладке Puppet Classes. Будут отображены все классы, находящиеся в выбранном на вкладке Host первом окружении Puppet.

С помощью группы хостов.

Группы хостов обычно соответствуют роли инфраструктуры, так как каждому хосту может быть назначена только одна группа хостов, и обычно большая часть его классов наследуется именно таким образом. Классы Puppet могут быть назначены путем редактирования группы хостов и выбора их на вкладке Puppet Classes.

Большая часть атрибутов группы хостов копируется на хост при его создании, однако связи с классами Puppet остаются унаследованными от группы хостов на протяжении его жизни. Любое изменение назначенных классов или параметров группы хостов будет влиять на любой хост с этой группой.

Атрибут окружения Puppet может отличаться на хосте и группе хостов, что означает, что классы Puppet, назначенные группе хостов, могут не отображаться в окружении Puppet хоста,. Любые классы Puppet, унаследованные от группы хостов, но отсутствующие в окружении хоста, не будут учитываться при создании данных ENC (YAML) Центр Управления. Проверьте раздел Configure > Puppet classes, чтобы убедиться, что классы доступны как в окружении группы хостов, так и в окружении хоста, если они отличаются.

Вы также можете массово назначить группу хостов нескольким хостам - выберите флажки требуемых хостов в списке хостов, а затем выберите Change Group в выпадающем меню Select Action в верхней части страницы.

Использование групп конфигурации.

Группа конфигурации предоставляет простой способ связывания множества классов Puppet с хостом или группой хостов. Обычно это используется для добавления определенного профиля приложения или стека в один шаг.

Чтобы создать группу конфигурации, перейдите в раздел Настройки > Config groups, нажмите "Создать группу конфигурации", введите имя и выберите желаемые классы Puppet. При редактировании хоста или группы хостов новую группу конфигурации можно добавить в верхней части вкладки Puppet Classes.

Группы конфигурации не являются специфичными для окружения, поэтому будут перечислены только те классы Puppet, которые находятся в окружении хоста при создании данных ENC (YAML). Любые классы, не перечисленные в окружении, будут пропущены.

Обратите внимание, что нельзя использовать замещение параметров класса с группой конфигурации, поскольку у хоста может быть много групп конфигурации без возможности определить их приоритет. Замещения должны быть выполнены на уровне группы хостов, хоста или других атрибутов.

Проверка результатов.

Чтобы увидеть точный список классов, которые будут переданы хосту с учётом Групп и Групп конфигурации, перейдите на страницу хоста и нажмите кнопку YAML. Вам будет показан точный YAML файл, отправляемый агенту Puppet - классы будут находиться в хэше "classes".

Параметры

Центр Управления позволяет передавать два типа параметров в Puppet: глобальные параметры и параметры классов.

Глобальные параметры доступны из любого манифеста и могут быть добавлены несколькими способами через Центр Управления. В языке Puppet глобальные параметры можно получить, используя `$:example` (предпочтительно) или `$example` для параметра с именем "example" в Центре Управления. При просмотре ENC (YAML) вывода из Центр Управления глобальный параметр будет представлен следующим образом:

```
parameters:  
  example: "foo bar"
```

Хосты наследуют свой список глобальных параметров из следующих источников, в порядке возрастающего приоритета:

1. Глобально определенные параметры в разделе Configure > Global parameters. Они применяются ко всем хостам.
2. Параметры на уровне организации в разделе Administer > Organizations > edit > Parameters.
3. Параметры на уровне местоположения в разделе Administer > Locations > edit > Parameters.
4. Параметры на уровне домена в разделе Infrastructure > Domains > edit > Parameters.
5. Параметры на уровне подсети в разделе Infrastructure > Subnets > edit > Parameters.
6. Параметры на уровне операционной системы в разделе Hosts > Operating systems > edit > Parameters.
7. Параметры на уровне группы хостов в разделе Configure > Host groups > edit > Parameters.
8. Параметры на уровне хоста в разделе Hosts > All hosts > edit > Parameters.

Последний (наиболее специфичный) уровень глобальных параметров применяется только к одному хосту. Перейдите к редактированию хоста и переключитесь на вкладку Parameters, и вы увидите все унаследованные параметры с предыдущих уровней. Здесь вы можете переопределить любые из предварительно определенных параметров или задать новые. Глобальные параметры поддерживают различные типы данных и валидацию в соответствии с выбранным типом. С поддержкой типов поиск по значению параметра больше не разрешен.

Проверка.

Чтобы увидеть, как Центр Управления передает параметры в Puppet, перейдите к хосту и нажмите кнопку YAML. Вам будет показан точный набор данных YAML, отправляемых на сервер Puppet - параметры будут находиться в хэше "parameters".

Параметризованные классы

Поддержка параметризованных классов позволяет обнаруживать, импортировать и предоставлять параметры напрямую классам, которые их поддерживают, через ENC (External Node Classifier).

Настройка.

По умолчанию поддержка параметризованных классов включена в Центр Управления. Вы можете проверить это в разделе Administer > Settings > Puppet и убедиться, что значение Parametrized_Classes_in_ENC установлено в true.

Настройки

Для настройки класса в Центр Управления выполните следующие шаги:

1. Нажмите на класс, с которым вы хотите работать.
2. Вторая вкладка называется "Параметр Smart Class". Нажмите на неё, чтобы открыть настройки параметров.
3. Слева отображается список возможных параметров, поддерживаемых классом.

Теперь давайте рассмотрим опции настройки параметра:

- Key (Ключ): Не редактируется и используется только для информации.
- Description (Описание): Текстовое поле для информационных заметок. Не передается в Puppet и не используется в других местах.
- Puppet Environments (Среды Puppet): Не редактируется и используется только для информации.
- Override (Переопределение) (важно): Если этот флажок не установлен, Центр Управления не будет пытаться управлять этой переменной, и она не будет передаваться в Puppet через ENC.
- Key type (Тип ключа): Тип данных, которые мы хотим передать. Обычно это строка, но поддерживаются и другие типы данных. Нет простого способа узнать, какой тип данных ожидает Puppet, поэтому вам придется ознакомиться с кодом/документацией конкретного модуля. Изменение типа требует соответствующего значения "Default Value" (Значение по умолчанию).
- Default Value (Значение по умолчанию): Это значение будет импортировано из Puppet, но если в Puppet используется наследование классов, вы можете получить неинформативное значение вроде "

```
KaTeX parse error: Expected '}', got 'EOF' at end of input: {
```

Центр Управления::params::user}". Это происходит потому, что Центр Управления не следует наследованию, поэтому вам нужно установить разумное значение по умолчанию.

- Omit (Пропустить): Если этот флажок установлен, параметр не будет включен в ENC, передаваемый в Puppet по умолчанию. Это полезно, если вы хотите использовать значение, указанное по умолчанию в Puppet в большинстве случаев, но хотите переопределить его только в определенных случаях, указанных в матчах.
- Hidden value (Скрытое значение): Определяет, должны ли значения параметра умного класса быть скрытыми в пользовательском интерфейсе.

Чтобы задать нужное вам значение для параметра, установите флажок Изменить и введите нужное вам значение. Однако, это изменяет значение переменной для всех хостов. Чтобы назначить индивидуальные значения для разных хостов, используйте сопоставители. Для этого перейдите к разделу "Указать конкретные сопоставители" внизу страницы.

Предположим, что любая машина в окружении Puppet "development" должна использовать значение "dev" вместо "production" для параметра "user". Добавьте "environment" в конец списка

сопоставителей, затем нажмите кнопку "Добавить сопоставитель" и заполните "Тип атрибута", ожидаемое значение атрибута (напр. имя хоста) и значение, которое нужно передавать при соответствии атрибута ожидаемому значению.

Переопределение параметра для хоста Если Центр Управления управляет значением параметра класса ("override = true"), вы также можете обновить переопределение значения для конкретного хоста непосредственно из настроек самого хоста. Для этого откройте страницу редактирования хоста, перейдите на вкладку "Параметры" и увидите переменную, область видимости класса и текущее значение. Затем вы можете переопределить значение для этого хоста.

Если значение скрыто, вы можете нажать кнопку "Раскрыть" для временного отображения значения при редактировании. Нажатие кнопки не изменит свойство скрытия параметра, оно просто отобразит значение для редактирования.

Фактически это другой способ назначить сопоставитель. Более удобный. Но если вы вернетесь на страницу класса Puppet, вы увидите, что Центр Управления добавил сопоставитель для этого хоста.

Та же кнопка переопределения доступна на вкладке "Параметры" для группы хостов. Для более сложной логики, такой как сопоставление на основе фактов, используйте страницу класса Puppet.

Дополнительные возможности Смарт-параметры классов основаны на технологии смарт-сопоставителей и имеют ряд дополнительных функций, таких как валидация и поддержка нескольких типов данных. Более подробную информацию об этих возможностях можно найти в разделе "Смарт-сопоставители".

Смарт-сопоставители

Технология смарт-сопоставления лежит в основе смарт-параметров классов и описана ниже.

Она предоставляет следующие возможности для каждого параметра:

Значение по умолчанию, которое может быть отправлено, если не найдено конкретного совпадения. Порядок приоритета для переопределений на основе атрибутов хоста или фактов. Список переопределений (сопоставителей). Указание типа данных, позволяющее передавать строки, целые числа и структуры данных нативно в Puppet. Опциональная проверка значений. Обработка значений с использованием шаблонов для создания динамического контента. Упорядочивание Переопределения обрабатываются в порядке приоритета, заданном в поле "Порядок" (от наиболее до наименее конкретного совпадения, первое совпадение побеждает, если включено слияние). Это список атрибутов хоста и имен фактов, которые будут проверяться для сопоставления с переопределениями. Если ни одно переопределение из этого списка не совпадает, используется значение по умолчанию.

Примеры атрибутов, которые могут быть указаны:

- fqdn - полное имя хоста («host.example.com»)

- `hostgroup` - полное имя группы хостов, включая родителей («Europe/Web servers»). Сопоставители на группы хостов могут наследоваться их дочерними группами, см. документацию по параметрам наследования в опциях конфигурации.
- `os` - имя и версия операционной системы («RedHat 6.4»)
- `domain` - доменное имя хоста («example.com»)
- `location` или `organization` - полное имя местоположения/организации, включая родителей («Company/Subsidiary»).

Сопоставители на местоположение/организацию могут наследоваться их дочерними местоположениями/организациями, см. документацию по параметрам наследования в опциях конфигурации. `is_virtual` - факт, предоставляемый Facter Порядок по умолчанию задается в разделе Администрирование > Настройки > Puppet > Путь по умолчанию для поиска переменных, и составляет "fqdn", "hostgroup", "os", "domain".

Слияние переменных.

При типе данных "кортеж" или "массив" отметка "Слияние переопределений" приведет к слиянию значений из каждого переопределения, которое совпадает (например, FQDN и домен).

Слияние выполняется "глубоко", поэтому вложенные хэши и массивы будут получать значения, а не полностью перезаписываться.

Опция "Слияние по умолчанию" добавляет значение по умолчанию как одно из значений для слияния, оно получит наименьший приоритет, поэтому одно из других значений может его перезаписать.

При типе данных "массив" опция "Исключить дубликаты" удалит повторяющиеся значения из результирующего массива.

Типы данных

Тип данных, который мы хотим передать в Puppet, можно установить в поле "Тип параметра". Обычно это строка, но также поддерживается множество других типов данных:

- `String` (Строка) - Все интерпретируется как строка.
- `Boolean` (Логическое значение) - Принимаются распространенные представления логических значений, включая `true`, `false`, `yes`, `no` и т.д.
- `Integer` (Целое число) - Только целые числа, могут быть отрицательными.
- `Real` (Действительное число) - Принимает любой числовой ввод.
- `Array` (Массив) - Допустимый ввод в формате JSON или YAML, который должен быть представлен массивом.
- `Hash` (Кортеж) - Допустимый ввод в формате JSON или YAML, который должен быть объектом/словарем/хэшем.
- `YAML` - Любой допустимый ввод в формате YAML.
- `JSON` - Любой допустимый ввод в формате JSON.

Нет простого способа определить, какой тип данных ожидается в манифесте Puppet, поэтому вам нужно прочитать код/документацию, которая идет вместе с конкретным модулем, чтобы узнать. Изменение поля типа требует соответствующего значения в поле "Значение по умолчанию".

Проверка ввода.

Раздел "Опциональный валидатор ввода" может использоваться для ограничения допустимых значений параметра. Важно отметить, что проверка применяется как при изменении со страницы редактирования хоста, так и при изменении со страницы редактирования классов Puppet.

Раздел валидации ввода скрыт по умолчанию, но его можно открыть, щелкнув на его заголовке.

При изменении типа параметра этот раздел автоматически развернется, чтобы изменить проверки в соответствии с новым типом.

Тип валидатора - выпадающий список типов данных. Этот тип применяется к следующему полю, валидатору. Правило валидатора - используется для принудительного соблюдения определенных значений параметров. См. примеры ниже.

Валидаторы строк

В настоящее время тип данных "строка" не может быть проверен - оставьте поле валидатора пустым, и все строки в переменной будут считаться допустимыми.

Валидаторы Regexp (регулярное выражение) / Список Путем ввода списка (разделенного запятыми, без пробелов) или регулярного выражения (без необходимости в разделителе) значение, которое будет присвоено параметру, будет проверяться по этому списку. Если значение не соответствует валидатору, будет возбуждена ошибка.

Шаблонные переменные Поскольку Центр Управления предлагает возможности шаблонизации, вы можете использовать предварительно существующие переменные, макросы и функции в ваших классах с параметрами. Это особенно полезно, если вам нужно передать строку в Puppet, но при этом встроить в строку информацию, специфичную для хоста, такую как его полное доменное имя (FQDN).

Ansible

Использование Ansible с Центром Управления позволяет настроить интеграцию и автоматизацию управления хостами. Вот несколько ключевых шагов и особенностей, о которых следует знать при работе с Ansible в Центр Управления:

- Роли могут быть импортированы из Агента, на котором включена функция 'Ansible'. Вы можете получить доступ к списку импортированных ролей через меню 'Настройка'. Выберите нужный источник для импорта из выпадающего меню.
- Импорт переменных: Аналогично ролям, переменные внутри ролей можно импортировать таким же способом. Перейдите в меню 'Настройка > Переменные Ansible', чтобы увидеть список уже импортированных переменных. При запуске нового импорта переменных Ansible,

если переменные принадлежат роли, которая еще не была импортирована, она будет автоматически импортирована вместе с переменными Ansible.

- Создание переменных в Центре Управления: Вы можете создавать переменные напрямую в Центр Управления. Перейдите в меню 'Настройка > Переменные Ansible' и нажмите кнопку 'Новая переменная Ansible' в верхней части таблицы переменных. Ключ и роль Ansible являются обязательными атрибутами. Переменные, созданные напрямую в Центр Управления, игнорируются при импорте переменных из прокси, и они не будут предлагаться для удаления.
- Выбор ролей для хоста: Роли Ansible могут быть назначены хосту на вкладке 'Роли Ansible' на странице редактирования хоста. Также возможно наследование ролей Ansible от группы хостов.

После развертывания хоста и его регистрации (с помощью `Центр Управления_url('built')` в шаблоне), Центр Управления выполнит первый запуск Ansible для настройки хоста в соответствии с применяемыми ролями. Поскольку Центр Управления не может знать, как долго потребуется, чтобы хост снова стал доступным после установки ОС и перезагрузки, Центр Управления устанавливает тайм-аут Ansible на 5 минут. Это можно настроить в меню 'Администрирование > Настройки', в параметре 'Тайм-аут после развертывания', или индивидуально для каждого хоста, переопределив эту настройку с помощью параметра.

Наследуемые роли автоматически добавляются при изменении группы хостов, и интерфейс выбора отключает прямое удаление таких элементов.

Запуск ролей Ansible

Чтобы запустить плейбук с выбранными ролями, нажмите кнопку 'Выполнить роли Ansible', и Центр Управления запустит плейбук в фоновом режиме. Это типовой плейбук, встроенный в Центр Управления, единственная функция которого - вызвать по очереди все роли, назначенные хосту. Роли должны быть составлены в соответствии с философией Ansible Galaxy и быть самодостаточными. В дальнейшем этот плейбук можно будет вызывать через интерфейс "Занести в расписание удаленное задание". Журналы выполнения плейбука можно найти в обычном журнале Центр Управления, обычно по пути `/var/log/Центр Управления/production.log`.

Альтернативно, можно запустить плейбук для всех ролей Ansible, применимых к хостам в группе хостов, нажав кнопку 'Выполнить роли' на странице групп хостов.

Если вы желаете выполнить произвольный плейбук, вам необходимо добавить такой плейбук как шаблон Ansible и сохранить, назначив имя.

Запуск плейбуков Ansible (Шаблоны заданий): Чтобы запустить плейбук:

- На странице хоста, нажмите 'Запланированное удаленное задание', выберите Шаблон задания и отправьте его.
- В списке хостов, выберите несколько хостов, затем нажмите 'Выбрать действие' -> 'Запланированное удаленное задание'.
- В списке Шаблонов заданий найдите нужный Шаблон задания и нажмите 'Запустить', затем отправьте его.

Переопределение переменных Ansible в плейбуках: Переменные Ansible ссылаются в плейбуках и шаблонах с использованием синтаксиса "{{ myvariable }}". Например:

```
template: src=foo.cfg.j2 dest={{ remote_install_path }}/foo.cfg
```

использует переменную `remote_install_path`. Вы можете использовать Центр Управления для переопределения этих переменных. Переменные будут переопределены только при запуске ролей из Центр Управления, так как Центр Управления отправляет инвентарь со словарем, содержащим имена переменных и их значения при выполнении ролей Ansible.

Есть несколько способов переопределить переменную Ansible:

- Импортируйте переменную, следуя инструкциям в разделе документации. После импорта переменной вы можете изменить ее значение по умолчанию.
- Еще один способ переопределить переменную - задать ее как входной параметр шаблона задания. Создайте Шаблон задания, содержащий плейбук с переменной {{ command }}.
- Вы можете переопределить эту переменную при каждом запуске Шаблона задания, задав переменную как входной параметр. Это имеет приоритет над параметрами хоста.

Salt

Создание группы хостов с использованием Salt Для объединения настроек предоставления услуг и конфигурации вы можете создать группу хостов с включенным Salt для управляемых хостов.

Процедура:

1. В веб-интерфейсе Центр Управления перейдите в раздел "Настроить" > "Группы хостов".
2. Нажмите "Создать группу хостов".
3. Перейдите на вкладку "Группа хостов" и выберите Salt-окружение и Salt-мастер.
4. Перейдите на вкладку "Salt States" и назначьте Salt-состояния вашей группе хостов.
5. Нажмите "Отправить" для сохранения группы хостов.

Дополнительную информацию смотрите в разделе "Создание группы хостов" в руководстве по управлению хостами.

Развертывание хостов-миньонов Разверните управляемые хосты, которые полностью предоставлены и настроены для использования Salt.

Необходимые предварительные условия для развертывания Salt-миньонов:

- Salt-мастер
- Salt-окружение

Процедура:

1. В веб-интерфейсе Центр Управления перейдите в раздел "Хосты" > "Создать хост".
2. Выберите ранее созданную группу хостов с включенным Salt.
3. Нажмите "Отправить" для развертывания хоста.

Проверка соединения между Salt-мастером и Salt-миньонами Проверьте соединение между вашим Salt-мастером и Salt-миньонами.

Процедура:

1. Подключитесь к своему Salt-мастеру с помощью SSH:

```
ssh root@salt-master.example.com
```

2. Проверьте доступность ваших Salt-миньонов:

```
salt "*" test.ping
```

3. Отобразите все Salt-грейны всех подключенных Salt-миньонов:

```
salt "*" grains.items
```

Salt-миньоны, управляемые через Центр Управления, связаны с Salt-мастером и Salt-окружением. Связанное Salt-окружение в Центр Управления должно совпадать с фактическим Salt-окружением из опции `file_roots` в файле `/etc/salt/master`. Вы можете настроить управляемые хосты с помощью Salt, как только они связаны с вашим сервером Центр Управления или прокси-сервером Агент, и установлен клиентское программное обеспечение Salt Minion.

Использование Salt Hammer CLI Вы можете использовать Hammer CLI для настройки управляемых хостов с использованием Salt. Запустите `hammer --help` для получения дополнительной информации.

Примеры.

Создание Salt-state:

```
hammer salt-state create \  
--name My_Salt_State
```

Просмотр информации о Salt-миньоне:

```
hammer salt-minion info \  
--name salt-minion.example.com
```

Добавление Salt-состояний к Salt-миньону:

```
hammer salt-minion update \  
--name salt-minion.example.com \  
--salt-states My_Salt_State
```

Импорт Salt-state

Salt-state настраивает различные части хоста, например, сервис или установку пакета. Вы можете импортировать Salt-state с вашего Salt-мастера в Центре Управления. Конфигурация Salt-мастера в этом руководстве использует Salt-окружение под названием `base`, которое включает Salt-состояния, хранящиеся в `/srv/salt/`.

Процедура:

1. В веб-интерфейсе Центр Управления перейдите в Configure > Salt States и нажмите Import from FQDN.
2. По желанию: нажмите Edit, чтобы назначить Salt-состояния для Salt-окружений.
3. По желанию: нажмите Delete, чтобы удалить Salt-состояние из Центр Управления. Это удаляет только Salt-состояние из Центр Управления, но не с диска вашего Salt-мастера.
4. Нажмите Submit, чтобы импортировать Salt-состояния.

После импорта Salt-состояний вы можете назначить их хостам или группам хостов. Salt применяет эти Salt-состояния ко всем назначенным хостам каждый раз при запуске state.highstate.

Просмотр ключей автоматической подписи Salt На странице Salt Keys перечислены хосты и их ключи Salt. Вы можете вручную принимать, отклонять или удалять ключи.

Используйте функцию Salt autosign для автоматического принятия запросов на подпись от управляемых хостов. По умолчанию управляемым хостам предоставляется ключ Salt во время создания хоста.

Примечание: Эта функция охватывает только процедуру аутентификации autosign через autosign.conf.

Процедура:

1. В веб-интерфейсе Центр Управления перейдите в Infrastructure > Smart Proxies.
2. Выберите Агент.
3. В выпадающем меню Actions нажмите Salt Keys.

Просмотр отчетов Salt.

Процедура: Чтобы просмотреть отчеты Salt, в веб-интерфейсе Центр Управления перейдите в Monitor > Config Management Reports.

Чтобы просмотреть отчеты Salt, связанные с отдельным хостом, в веб-интерфейсе Центр Управления перейдите в Hosts > All Hosts и выберите один хост.

Переменные Salt

Вы можете настраивать переменные Salt в Центр Управления. Настроенные значения доступны в качестве данных Salt Pillar.

Просмотр параметров ENC.

Вы можете использовать Центр Управления в качестве внешнего классификатора узлов для Salt. Нажмите Salt ENC на странице обзора хоста, чтобы просмотреть назначенные Salt-состояния. Это покажет список параметров, которые доступны для использования в Salt в качестве данных Salt Pillar.

Чтобы проверить, какие параметры действительно доступны на стороне Salt, выполните

следующую процедуру.

Процедура: Подключитесь к вашему Salt-мастеру по SSH:

```
ssh root@salt-master.example.com
```

Просмотрите доступные параметры ENC:

```
salt '*' pillar.items
```

По желанию: обновите данные Salt Pillar, если какой-то параметр отсутствует:

```
salt '*' saltutil.refresh_pillar
```

Выполнение Salt.

Вы можете выполнять произвольные функции Salt с использованием функций удаленного выполнения Центр Управления.

Наиболее часто вы можете выполнять salt.highstate на одном или нескольких Salt-минионах. Это применяет все соответствующие Salt-состояния на ваши управляемые хосты.

Процедура: В веб-интерфейсе Центр Управления перейдите в Hosts > All Hosts.

Выберите один или несколько хостов.

Нажмите Run Salt в выпадающем меню действий.

Кроме того, вы можете нажать Run Salt в выпадающем меню Schedule Remote Job на странице обзора хоста.

Запуск задания удаленного выполнения перенесет вас на соответствующую страницу обзора задания.

Если вы выбрали хост из списка, вы можете видеть вывод удаленного задания в реальном времени.

Чтобы просмотреть удаленные задания, в веб-интерфейсе Центр Управления перейдите в Monitor > Jobs.

Запуск state.highstate генерирует отчет Salt. Обратите внимание, что отчеты загружаются в Центр Управления каждые десять минут.

Вы можете использовать функции удаленного выполнения Центр Управления для выполнения произвольных функций Salt. Это включает опциональный тестовый режим, который позволяет узнать, что произошло бы без фактического изменения.

Процедура:

В веб-интерфейсе Центр Управления перейдите в Hosts > All Hosts

Выберите хост.

Нажмите Schedule Remote Job.

В поле Job category выберите Salt-Call из раскрывающегося меню.

В поле Job template выберите Salt Run function из раскрывающегося меню.

В поле Function введите имя функции, которую вы хотите запустить, например, pillar.items или test.ping.

По желанию: Нажмите Display advanced fields и включите режим тестирования (test-run).
Нажмите Submit, чтобы запустить задание.

Вы также можете планировать удаленные задания или выполнять их периодически.
Дополнительную информацию см. в разделе Configuring and Setting up Remote Jobs в руководстве Managing Hosts.

В качестве альтернативы вы можете определить периодические действия с помощью стандартных методов Salt. Например, вы можете запланировать выполнение state.highstate каждый час на отдельных Salt-минионах, расширив файл /etc/salt/minion:

```
schedule:  
  highstate:  
    function: state.highstate  
    minutes: 60
```

Пример использования Salt State для управления файлом /etc/motd на Salt Minions, включая необходимые шаги в Центр Управления:

Процедура:

1. Создайте глобальный параметр с названием "vendor_name" и значением "Центр Управления".
2. Добавьте новый Salt State с названием "motd" на ваш Salt Master.
3. Создайте директорию /srv/salt/motd/:
4. Создайте файл /srv/salt/motd/init.sls в качестве файла Salt State:

```
/etc/motd:  
  file.managed:  
    - user: root  
    - group: root  
    - mode: 0644  
    - source: salt://motd/motd.template  
    - template: jinja
```

5. Создайте файл /srv/salt/motd/motd.template в качестве шаблона, на который ссылается файл Salt State:

```
Welcome to {{ grains['fqdn'] }} Powered by {{ salt['pillar.get']  
( 'vendor_name' ) }}
```

6. Импортируйте Salt State "motd" в Центр Управления.
7. Убедитесь, что Salt имеет доступ к параметру "vendor_name", выполнив одну из следующих команд на вашем Salt Master:

```
salt '*' pillar.items | grep -A 1 vendor_name  
salt '*' pillar.get vendor_name
```

Если вывод не включает значение параметра "vendor_name", необходимо сначала обновить данные Salt Pillar:

```
salt '*' saltutil.refresh_pillar
```

8. Добавьте Salt State "motd" на ваши Salt Minions или в группу хостов.
9. Примените Salt State, выполнив команду state.highstate.
10. По желанию, проверьте содержимое файла /etc/motd на Salt Minion.

Веб-интерфейс

Роли и права доступа

Доступ пользователя к функциям Центр Управления ограничивается разрешениями, которые ему предоставляются. Эти разрешения также используются для ограничения набора хостов, групп хостов и других ресурсов, к которым пользователь может получить доступ и изменять.

Примечание: пользователь с включенной глобальной административной ролью не ограничен системой авторизации. Это значение по умолчанию для установок, в которых в файле config/settings.yml не указано значение :login:true.

Пользователь, авторизовавшийся в системе, будет наделен ролью "Default role", а также одной или несколькими дополнительными ролями. Права доступа и фильтры, связанные с этими ролями, объединяются и определяют конечный набор разрешений.

Роли могут администрироваться пользователями с правами администратора или обычными пользователями с разрешением "edit_roles". Чтобы добавить новые фильтры и разрешения в роль, обычным пользователям должно быть предоставлено разрешение "create_filters".

Роли Они могут быть созданы, удалены и отредактированы на странице "Roles". Каждая роль содержит фильтры разрешений, которые определяют разрешенные действия для определенного ресурса. После создания роли вы можете связать ее с одним или несколькими пользователями и группами пользователей.

Существует одна встроенная системная роль, "Default role". Это набор разрешений, который будет предоставлен каждому пользователю, в дополнение к любым другим ролям, которыми он обладает. Центр Управления предоставляет вам набор ролей по умолчанию. Эти роли могут быть назначены пользователям, но не могут быть изменены. Они служат в качестве разумного набора значений по умолчанию и быстрой отправной точки. Если вы хотите создать пользовательскую роль на основе одной из этих ролей, вы можете ее клонировать и изменить клон.

Роли также могут быть связаны с местоположениями (Locations) или организациями (Organizations), если это разрешено. В отличие от других объектов, это не означает, что роли будут доступны только в определенной области. Роли всегда являются глобальными для всей системы Центр Управления. Связь означает, что фильтры такой роли ограничены определенной организацией или местоположением. Представьте, что вы хотите создать роль, представляющую администратора организации А. Вы можете клонировать существующую роль администратора организации и связать ее с организацией А. Если вы позже назначите эту роль некоторым пользователям, они получат все разрешения администратора, но только на ресурсы организации А. Обратите внимание, что некоторые ресурсы не могут быть ограничены организациями или

местоположениями. Фильтры для таких ресурсов предоставляют глобальные разрешения. Встроенная роль администратора организации (Organization admin) похожа на роль менеджера (Manager). Обе роли автоматически расширяются разрешениями, введенными в новых версиях Центр Управления, а также разрешениями, введенными плагинами. Разница заключается в том, что роль администратора организации не содержит разрешений для управления организациями, только для просмотра. Поскольку администратору организации обычно не требуется создавать или изменять другие организации, роль администратора организации лучше подходит для этого сценария.

Роль системного администратора (System admin) является встроенной ролью с очень мощными возможностями. Цель этой роли - настроить среду для использования другими пользователями. Она может создавать организации/местоположения, но не имеет доступа к ресурсам внутри них. Системный администратор может создавать новых пользователей, назначать их в организации/местоположения и добавлять роли пользователям. Системный администратор может просматривать и изменять настройки. Но самое главное, пользователи с этой ролью могут делегировать роли, которыми они сами не обладают. Пользователи, имеющие эту роль, могут потенциально выйти из нее, создав нового пользователя с ролями/разрешениями, которых они сами не имеют как системные администраторы, и войти в систему от имени нового пользователя. Поэтому только доверенным пользователям следует разрешать иметь эту роль.

Фильтры

Фильтры определяются в контексте роли, щелкнув по ссылке "фильтры и разрешения". Фильтр позволяет пользователю выбрать ресурс (хосты, группы хостов и т. д.) и разрешения, которые должны быть предоставлены для этого ресурса. После создания фильтра пользователи, которым назначена роль с этим фильтром, будут иметь разрешения для указанного ресурса, указанные в фильтре.

Если фильтр помечен как "Неограниченный?", разрешения, созданные в этом фильтре, будут применяться ко всем объектам выбранного ресурса. Например, если ресурс - это хост, и разрешения - "просмотр" и "индекс", а "Неограниченный?" отмечен, пользователи, у которых есть роль с этим фильтром, смогут "просматривать" и "индексировать" все хосты в системе. Когда "Неограниченный?" не отмечен, будет включено текстовое поле, в котором можно определить более детальное фильтрование. Вы можете написать поисковый запрос, и разрешения в этом фильтре будут применяться только к результатам этого запроса. Пример запроса для ресурса "Хост" может быть "os = RedHat". В этом случае разрешения в этом фильтре будут применяться только к хостам, у которых операционная система установлена на Red Hat. Вы можете проверить свои поисковые запросы на странице индекса вашего ресурса, в данном случае это будет '/hosts'.

Некоторые примеры запросов для ресурса "Хост":

Принадлежность и членство в домене: 'owner_id = 95 and domain = localdomain' - Будет применять разрешения к хостам, принадлежащим пользователю с идентификатором 95 и находящимся в домене 'localdomain'.

Членство в ресурсе вычисления: 'compute_resource = Openstack' - Будет применять разрешения к хостам, развернутым на ресурсе вычисления Openstack.

Фильтрация фактов: 'facts.alarmlevel = high' - Будет применять разрешения к хостам с фактом

'alarmlevel' со значением 'high'. Поскольку факт генерируется только во время выполнения Puppet, этот фильтр будет относиться только к машинам, которые были созданы, и, следовательно, нельзя использовать для ограничения создания машин.

Эти наборы запросов можно объединять, добавляя их вместе, или фильтры можно использовать для ограничения выбранного ресурса до все меньшего подмножества общего. Можно представлять их как операции над множествами.

Как уже упоминалось, роль может быть назначена организациям и местоположениям. В таком случае все фильтры для ресурсов, поддерживающих такое разделение, автоматически применяют те же организации и местоположения. Если вы хотите объединить фильтры с разными назначениями организаций или местоположений, вы можете использовать флажок "Переопределить". Если он отмечен, вы можете изменить организации и местоположения для фильтра. Если вы снимите этот флажок, фильтр начнет наследовать организации и местоположения роли после повторной отправки. Если вы хотите сбросить все фильтры роли и начать наследование, вы можете использовать кнопку "Отключить все переопределения фильтров" на вкладке "Фильтры" роли.

Мы рекомендуем управлять связью организаций и местоположений на уровне роли, чтобы сохранить простоту и ясность настройки.

Примечание: Если флажок "Администратор" установлен для пользователя, фильтрация не будет действовать.

Разрешения Они определяют операции, которые разрешено выполнять с ресурсами, на которые они ссылается. Для нескольких простых элементов, таких как закладки, это работает ожидаемым образом - оно предоставляет разрешение для всех закладок. Но для большинства ресурсов, таких как хосты, над которыми пользователь может выполнять операции, существует дополнительный уровень безопасности, называемый фильтрацией.

При редактировании фильтра есть поле поиска внизу, которое сужает область разрешений, предоставляемых для подмножества объектов ресурса. Большинство типов разрешений поддерживают это поле поиска, но есть некоторые исключения. Разрешение для создания объектов не может быть ограничено поисковым запросом, потому что объект не существует во время создания. Поэтому пользователю предоставляется разрешение на создание, если они связаны с любым фильтром, содержащим это разрешение (с ограничением поисковым запросом или нет).

В следующей таблице перечислены некоторые разрешения и их воздействие:

Разрешение	Описание
Разрешения для архитектур, провайдеров аутентификации, сред, внешних переменных, общих параметров, носителей, моделей, операционных систем, таблиц разделов, классов Puppet и групп пользователей view	Разрешено просматривать этот тип объекта при их перечислении на странице индекса create
edit	Разрешено создавать этот тип объекта
destroy	Разрешено редактировать этот тип объекта
view	Разрешено удалять этот тип объекта
view	Разрешения для доменов
create	Разрешено просматривать список доменов при просмотре страницы индекса create
edit	Разрешено создавать новый домен, а также создавать параметры домена
destroy	Разрешено редактировать домен и его параметры. Если у пользователя активна фильтрация доменов в его профиле, то только эти домены будут доступны для редактирования
destroy	Разрешено удалять домен и его параметры. Если у пользователя активна фильтрация доменов в его профиле, то только эти домены будут доступны для удаления
	Разрешения для

групп хостов view Разрешено просматривать список групп хостов при просмотре страницы индекса create Разрешено создавать новую группу хостов, а также создавать параметры группы хостов edit Разрешено редактировать группу хостов и ее параметры. Если у пользователя активна фильтрация групп хостов в его профиле, то только эти группы хостов будут доступны для редактирования destroy Разрешено удалять группу хостов и ее параметры. Если у пользователя активна фильтрация групп хостов в его профиле, то только эти группы хостов будут доступны для удаления Разрешения для хостов view Разрешено просматривать список хостов при просмотре страницы индекса. Этот список может быть ограничен фильтрами хостов пользователя create Разрешено создавать новый хост. Эта операция может быть ограничена фильтрами хостов пользователя edit Разрешено редактировать хост. Эта операция может быть ограничена фильтрами хостов пользователя destroy Разрешено удалять хост. Эта операция может быть ограничена фильтрами хостов пользователя Разрешения для пользователей view Разрешено просматривать список пользователей при просмотре страницы индекса. Пользователь всегда сможет видеть свою учетную запись, даже если у него нет этого разрешения create Разрешено создавать нового пользователя edit Разрешено редактировать существующих пользователей. Пользователь всегда сможет редактировать свои основные настройки учетной записи и пароль destroy Разрешено удалять пользователей из системы

Аудит

Центр Управления поддерживает аудит практически всех изменений, которые происходят в системе, как через пользовательский интерфейс, так и через API. Аудит проводится на уровне пользователя и, следовательно, не работает, если параметр `:login:` установлен в значение `false`, поскольку все аудиты будут выполняться от имени пользователя "admin".

Основной вид Перейдите на вкладку "Аудит", чтобы увидеть просмотр изменений. В этом виде можно фильтровать изменения по типу или по объекту, который был изменен (например, можно выполнить поиск по имени хоста, чтобы увидеть все изменения, связанные с этим хостом). Вы также получаете родительский объект - таким образом, если был изменен параметр, вы можете увидеть, к какому хосту/группе принадлежит этот параметр. Дата и время изменения, а также имя пользователя, который выполнил изменение, будут указаны.

Расширенные аудиты для шаблонов При изменении шаблонов также сохраняется разница между версиями и имеется возможность откатиться к предыдущей версии шаблона.

Автоматическое удаление старых аудитов Вероятно, вы захотите удалить старые аудиты через некоторое время. Для этого есть планировщик задач (`cronjob`), который запускает задачу `rake audits:expire`. По умолчанию не задано количество дней, на которое следует хранить аудиты, потому что у каждого пользователя могут быть свои требования. Однако вы можете настроить количество дней для хранения в настройках Центр Управления (Администрирование -> Настройки -> Общие (вкладка) -> Интервал сохранения аудитов) в соответствии с вашими потребностями. Вы можете оставить значение пустым, чтобы не удалять старые аудиты. Также эту задачу можно выполнить вручную по необходимости. Запустите команду Центр Управления-`rake audits:expire`. Если параметр `days` не указан, задача пытается использовать значение, настроенное в настройках. Если это значение также пусто, задача завершается, и аудиты не удаляются.

Доступные параметры:

days => количество дней для хранения аудитов (по умолчанию 90) Примеры:

Удаление всех аудитов, старше 7 дней: Центр Управления-rake audits:expire days=7

Анонимизация старых аудитов Анонимизация очищает ссылки на учетные записи пользователей и их IP-адреса, но сохраняет все остальные данные аудита в базе данных.

Вместо удаления записей аудита вы можете выполнить анонимизацию старых аудитов. Для этого используйте задачу Центр Управления-rake audits:anonymize аналогично audits:expire.

Пример:

Анонимизация всех аудитов, старше 7 дней: Центр Управления-rake audits:anonymize days=7

Организации и местоположения Аудиты наследуют организации и местоположения ресурсов, для которых они были созданы. Представьте, что у вас есть подсеть, назначенная организации А. При каждом изменении этой подсети аудит будет виден только в организации А. Когда вы позже добавляете эту подсеть в организацию В, новые аудиты появятся как в организации А, так и в организации В. Все ранее созданные аудиты остаются неизменными. Ресурсы, которым нельзя назначить организацию или местоположение, всегда будут видны во всех организациях и местоположениях, что означает, что изменение затронуло все организации и местоположения.

История аудита Аудируемые ресурсы могут меняться со временем, например, появляются новые атрибуты. Также меняются определения аудита, например, начинают отслеживаться новые ассоциации в новой версии Центр Управления. Все эти изменения применяются только к новым созданным аудитам, старые аудиты не могут быть обновлены и всегда будут содержать только данные, известные на момент их создания. Начиная с версии Центр Управления 1.20, аудиты ограничены организациями и местоположениями. Все ранее созданные аудиты остаются непривязанными, что означает, что их видит только администратор.

Поиск

В каждой странице Центра Управления есть своя функция поиска, которая основана на управляемых ею ресурсах и позволяет выполнять поиск по атрибутам ресурсов в списке или ресурсам, с которыми они связаны. В поле поиска также доступна мощная автозаполнение для создания запросов поиска и свободный текстовый поиск на многих страницах. Функцию поиска также можно использовать в API при перечислении ресурсов, подробнее см. в разделе Настройка JSON-ответов.

Общее использование Поиск осуществляется с помощью запросов вида «поле = значение» или свободного текста, которые могут быть объединены с помощью логических операторов (и, или, не) и скобок для обработки более сложной логики. Несколько примеров:

name = client.example.com в списке хостов покажет хост(ы) с именем client.example.com hostgroup = "Web servers" and domain != lon.example.com покажет хосты в группе хостов Web servers, но не в домене lon.example.com Web servers покажет все хосты с этим текстом где-либо, например, в имени группы хостов или в поле комментария Доступные поля зависят от типа ресурса, по которому производится поиск, и имена атрибутов варьируются в зависимости от контекста. Пожелания по добавлению дополнительных полей для поиска приветствуются и могут быть поданы в категорию "Поиск" в системе отслеживания ошибок.

Поисковый движок предоставляется библиотекой scored_search, которая преобразует запросы

поиска непосредственно в SQL-запросы. Документация по языку запросов предоставляет более полную спецификацию доступного синтаксиса.

Закладки Центр Управления поддерживает возможность создания закладок для поиска, позволяя пользователям быстро переходить к predetermined условиям поиска. Доступные закладки можно выбрать из выпадающего меню справа от поля поиска или управлять ими в разделе Администрирование > Закладки.

Некоторые закладки предоставляются по умолчанию, например, для поиска активных или неактивных хостов или для просмотра только отчетов с событиями.

Чтобы сохранить запрос, используйте выпадающее меню справа от поля поиска и нажмите "Добавить закладку для этого поиска". При сохранении закладки можно пометить как публичную, чтобы все другие пользователи могли ее видеть и использовать.

Свободный текстовый поиск Если вы игнорируете автозаполнение и просто вводите текст в поле поиска, Центр Управления попытается выполнить поиск этого текста в нескольких полях.

Например, если вы просто вводите 12 в поле поиска хостов, результаты будут включать все хосты с 12 в их IP-адресе, MAC-адресе или имени. В целом поля, используемые для свободного текстового поиска, минимизируются с целью повышения производительности и точности.

Желательно выполнять поиск с использованием конкретного поля, например, при поиске по IP-адресу используйте ip ~ 12 вместо 12.

Поиск значений "присутствует/пусто" Оператор "has" соответствует значениям, которые присутствуют, например, для поиска хостов, находящихся на вычислительном ресурсе, используйте has compute_resource.

Аналогично это можно отрицать, так что для поиска хостов без групп хостов вы можете использовать not has hostgroup.

Регистрозависимость При использовании операторов = и != точное сопоставление будет возвращать совпадения с учетом регистра. При использовании операторов ~ (подобно) и !~ (не подобно) сопоставление не зависит от регистра.

Цитирование В поисковых запросах пробелы используются в качестве разделителей. Вот несколько примеров того, как будет интерпретироваться запрос:

description ~ "created successfully": показывать все уведомления, содержащие "created successfully" description ~ created successfully: показывать все уведомления, содержащие "created" и хотя бы одно из его текстовых полей содержит "successfully" description !~ created successfully: показывать все уведомления, не содержащие "created" и хотя бы одно из его текстовых полей содержит "successfully" Во втором и третьем примере "successfully" - это дополнительный термин, который интерпретируется как свободный текстовый поиск.

Маски ('', '%', '*') Операторы поиска ~ и !~ переводятся в соответствующие SQL-запросы LIKE и NOT LIKE, которые поддерживают два основных символа-маски: _ и %.

_ - это символ-маска для замены одного символа. Например, поиск name ~ fo совпадет и с "foo", и с "for".

Символ-маска % и * заменит ноль или более символов. Например, поиск name ~ corp% будет соответствовать и "corp", и "corporation". Более распространенный символ-маска '*', хотя он не является символом-маской SQL, может быть использован вместо него.

Когда обрабатывается поиск с использованием операторов ~ или !~, если не используется символ-маска, автоматически добавляется символ-маска '%' в начале и конце значения, чтобы

оно по умолчанию соответствовало в любом месте внутри строки. Например, поиск `name ~ foo` эквивалентен `name ~ %foo%`, а поиск `name ~ foo%` будет соответствовать только "foo" в начале значения.

Синтаксис запроса поиска даты и времени В поисковых запросах принимаются различные форматы даты и времени. Вот несколько примеров:

"30 minutes ago", "1 hour ago", "2 hours ago", Today, Yesterday "3 weeks ago", "1 month ago", "6 days ago", "July 10,2011" Дата может иметь разные разделители. "10-July-2011" будет интерпретироваться так же, как "10/July/2010" или "10 July 2011". Имена месяцев могут быть полным английским названием или трехбуквенным сокращением, например, "Jan" будет интерпретироваться как "January". Также принимаются многие другие форматы, однако не рекомендуется использовать двусмысленные форматы, такие как "3/4/2011".

Допустимые операторы даты и времени - '=', '<' и '>', которые интерпретируются как 'в', 'до' и 'после' соответственно. Вот как интерпретируется условие поиска:

Правая часть условия даты и времени разбирается и преобразуется в определенное дату и время. "30 minutes ago" преобразуется в "сейчас - 30 минут".

`last_report > "2011-07-01 12:57:18 EDT"` должно быть прочитано как "создан после этого времени" Аналогично, `last_report > "30 minutes ago"` должно быть прочитано как "создано после 30 минут назад", а не "создано более 30 минут назад" Поиск, например, `installed_at = Yesterday`, переводится в запрос, который будет искать значения, равные дате "вчера", а не "вчера в это же время". Время по умолчанию будет 00:00:00.

Это некоторые основные принципы использования поиска в Центр Управления. У Центр Управления есть богатый набор функций поиска, которые могут быть использованы для точного и гибкого поиска ресурсов.

Управление пользователями Центр Управления предоставляет возможность управления пользователями, которые взаимодействуют с хостами.

SSH-ключи Каждому пользователю Центр Управления можно назначить несколько SSH-ключей при редактировании профиля пользователя. Сами по себе эти ключи не имеют особого назначения, но их можно использовать в шаблонах развёртывания и получать доступ к ним через данные ENC. Они облегчают управление пользователями и ключами для входа по SSH на хостах без необходимости использования LDAP. Если вы хотите, чтобы пользователи могли входить на хосты, используя данные, предоставленные в Центр Управления, вам необходимо включить фрагмент `create_users` в вашем шаблоне развёртывания. Для синхронизации данных пользователей между Центр Управления и ваших хостов также доступен модуль `Puppet`.

Шаблоны

В этой главе мы опишем, как настроить Агент для обслуживания шаблонов развёртывания. Агент способен проксировать запросы на шаблоны от хостов в изолированных сетях к серверу Центр Управления, если агент также обрабатывает TFTP.

Эта функция зависит от правильно настроенного `trusted_proxies` на сервере Центр Управления. В общем случае вы хотите, чтобы шаблоны были доступны как по протоколу HTTP, так и по HTTPS. При включении HTTP на агенте убедитесь, что конфигурации других модулей в файлах `/etc/Центр Управления-proxy/settings.d/* .yml` защищены, установив `:enabled:` в `https` вместо `true`.

Убедитесь, что значение `Центр Управления_url` в файле `/etc/Центр Управления-proxy/settings.yaml` указывает на ваш экземпляр Центр Управления, и что ваш агент слушает HTTP, раскомментировав `http_port`. Теперь настройте файл `/etc/Центр Управления-proxy/settings.d/templates.yml`:

```
:enabled: true :template_url: http://smart-proxy.example.com:8000
```

После выполнения вышеперечисленных шагов перезапустите сервис Центр Управления-proxy и обновите функции на вашем сервере Центр Управления.

Функция шаблонов будет использоваться автоматически: любой хост, который использует этот агент для TFTP, также будет использовать агент для получения своих шаблонов.

Журналы

Модуль логов агента предоставляет API для получения недавно зарегистрированных сообщений и информации о неудачных модулях. Это будет отображаться в Центр Управления на страницах агента, когда модуль будет включен.

Количество логов контролируется основными настройками журналирования агента, подробно описанными в настройках агента.

Управление электронной почтой

Центр Управления также может отправлять различные уведомления по электронной почте в соответствии с событиями или регулярным расписанием. Плагины также могут расширять эту функциональность, предоставляя свои собственные сводки и уведомления.

Для отправки электронной почты требуется настроенный сервер SMTP или локальный MTA (например, `sendmail`), который настраивается в разделе `Administer > Settings > Email` в соответствии с настройками конфигурации.

Регулярные электронные сообщения отправляются с помощью задач `rake` (`reports:daily`, `reports:weekly`, `reports:monthly`), которые запускаются из `cron`-заданий и настраиваются в файле `/etc/cron.d/Центр Управления`.

Параметры электронной почты

Пользователи

Электронные сообщения отправляются отдельным пользователям, зарегистрированным в Центр Управления, на адрес электронной почты, указанный в их учетной записи. Пользователи могут изменить адрес электронной почты, щелкнув на своем имени в правом верхнем углу веб-страницы и выбрав "Моя учетная запись".

Чтобы изменить настройки подписки на сообщения для отдельного пользователя, в разделе "Параметры почты" в учетной записи пользователя перечислены все доступные типы сообщений и частота, с которой каждое сообщение должно быть получено. Также имеется глобальный флажок для отключения всех сообщений электронной почты от Центр Управления.

Уведомления на основе событий могут быть включены или отключены, и они отправляются из Центр Управления в момент возникновения события. Запланированные уведомления могут быть отправлены ежедневно, еженедельно или ежемесячно.

Хосты

Уведомления, связанные с хостами, могут быть отключены для каждого хоста по отдельности, что полезно, когда ожидаются ошибки. На вкладке "Дополнительная информация" хоста снимите флажок "Включено", чтобы отключить уведомления и исключить хост из отчетов. Включение и отключение уведомлений также можно сделать из списка хостов, используя флажки и выбирая пункт "Включить/Отключить уведомления" из раскрывающегося меню "Выбрать действие". Уведомления о событиях для хоста отправляются владельцу хоста. Владелец выбирается на вкладке "Дополнительная информация" хоста и может быть как отдельным пользователем, так и группой пользователей. Если выбрана группа пользователей, все члены группы, подписанные на данный тип электронной почты, получают уведомление.

Уведомления аккаунта

Приветственное письмо нового аккаунта Если включена опция `send_welcome_email` (Configuration Options), владельцы новых учетных записей получают электронное письмо с указанием своего имени пользователя и ссылкой на Центр Управления.

Уведомления о хосте

Завершение сборки

Когда хост завершает процесс сборки, либо уведомляя Центр Управления о завершении по запросу в конце автоматической установки, либо после удаленного выполнения сценария Центр Управления, владельцы хоста получают уведомление по электронной почте.

Ошибка Puppet

Когда получено отчетное сообщение от Puppet, которое помещает хост в состояние ошибки (красное состояние), владельцы хоста получают соответствующее уведомление по электронной почте.

Запланированные электронные письма

Сводка аудита

Регулярное письмо со сводкой всех изменений объектов в Центр Управления, которые вызвали аудиторские события (см. Аудит), включая пользователя, совершившего изменение, время изменения и ссылку для получения дополнительной информации.

Сводка Puppet Регулярный обзор всех хостов, к которым у пользователя есть доступ, и их состояния Puppet. Включает количество событий Puppet за период отчета, таких как примененные, пропущенные и неудачные ресурсы.

Мониторинг

Для мониторинга вашей инфраструктуры полезны статусы хостов. В Центр Управления каждый хост имеет глобальный статус, который указывает, какие хосты требуют внимания. Каждый хост также имеет подстатусы, которые представляют статус определенной функции. При любом изменении подстатуса глобальный статус пересчитывается, и результат определяется на основе статусов всех подстатусов.

Глобальный статус

Глобальный статус представляет общий статус конкретного хоста. Статус может иметь одно из трех возможных значений: OK (OK), Warning (Предупреждение) или Error (Ошибка).

OK означает, что никакие ошибки не были сообщены ни одним из подстатусов. Он обозначается зеленым цветом.

Warning (Предупреждение) подразумевает, что пользователь должен проверить статус, поскольку, хотя ошибок не было обнаружено, некоторый подстатус вызвал предупреждение. Хорошим примером может быть отсутствие отчетов Puppet для хоста, хотя хост настроен на отправку отчетов Puppet. Поэтому он выделяется желтым цветом.

Последнее возможное значение - Error (Ошибка), что указывает на то, что некоторый подстатус сообщает о сбое. Например, это может означать, что выполнение Puppet содержит неудачные ресурсы. Очевидно, это что-то, что следует исправить, и пользователь получает предупреждение красным цветом.

Вы можете найти глобальный статус на странице обзора хостов, отображаемый в виде небольшой иконки рядом с именем хоста соответствующего цвета. Наведение указателя мыши на иконку позволяет отобразить всплывающую подсказку с информацией о подстатусах, чтобы быстро получить дополнительные сведения.

Подстатусы

Подстатус мониторит только часть возможностей хоста. В настоящее время Центр Управления предоставляет только два подстатуса: подстатус сборки и подстатус конфигурации. Не все подстатусы являются актуальными для всех хостов, поэтому конфигурация рассматривается только в том случае, если хост использует систему управления конфигурацией, например, связан с каким-либо прокси-сервером Puppet. Подстатус сборки актуален для управляемых хостов и когда Центр Управления работает в автоматическом режиме.

Вы можете увидеть глобальный статус хоста со всеми подстатусами на странице деталей хоста в таблице свойств. Обратите внимание, что плагины могут добавлять дополнительные подстатусы. Все статусы

Каждый подстатус может определить свой набор возможных значений, которые отображаются в трех глобальных статусах. Подстатус сборки имеет два возможных значения - pending (ожидает) и built (построенные), которые оба отображаются как глобальный статус OK. Статус конфигурации более сложный, и его возможные значения и отображения описаны в таблице ниже.

Значение статуса Отображение Описание Ошибка Error Ошибка при конфигурации, например, не удалось установить некоторый пакет во время выполнения Puppet Не синхронизирован Предупреждение Отчет о конфигурации не был получен в ожидаемый интервал, основанный на

параметре `outofsync_interval` Нет отчетов Предупреждение / ОК Когда нет отчетов, но хост использует систему управления конфигурацией (например, связан с прокси-сервером Puppet) или параметр `always_show_configuration_status` установлен в `true`, он отображается как предупреждение. В противном случае отображается как ОК. Активный ОК Во время последнего выполнения Puppet были применены некоторые ресурсы Ожидающий ОК Во время последнего выполнения Puppet некоторые ресурсы были бы применены, но Puppet был настроен на режим `noop` Нет изменений ОК Во время последнего выполнения Puppet ничего не изменилось 1 Отчеты идентифицируются по источнику и могут иметь разные интервалы на основе этого источника. Например, отчеты Puppet будут иметь источник 'Puppet' и интервал, установленный параметром `puppet_interval`.

Поиск по статусам

Вы можете искать хосты по глобальному статусу. Вот некоторые примеры:

поиск всех хостов со статусом ОК

```
global_status = ok
```

поиск всех хостов, требующих внимания

```
global_status = error or global_status = warning
```

Чтобы искать хосты на основе статуса конфигурации, вы можете искать по последним метрикам отчетов, например, так:

найти хосты, у которых есть хотя бы один ожидающий ресурс

```
status.pending > 0
```

найти хосты, которые перезапустили некоторый сервис во время последнего выполнения Puppet

```
status.restarted > 0
```

найти хосты с интересным последним выполнением Puppet (что-то произошло)

```
status.interesting = true
```

Отчеты

Центр Управления предоставляет возможности для создания отчетов. Он использует тот же движок шаблонов, что и шаблоны развёртывания и таблицы разделов. Список макросов, которые можно использовать в этом контексте, немного отличается, так как отчеты обычно отображаются для конкретного хоста. Отчеты не накладывают никаких ограничений на формат вывода, так как форматирование является частью самого шаблона, единственное требование - чтобы результат вывода был в текстовой форме.

Шаблоны отчетов

Существующие шаблоны отчетов можно найти в разделе "Монитор" -> "Шаблоны отчетов". Центр Управления поставляется с несколькими шаблонами отчетов по умолчанию, которые заблокированы. Как и в случае с другими шаблонами, мы не рекомендуем их разблокировку и изменение, так как они автоматически обновляются при обновлениях. Если требуется изменение шаблона по умолчанию, рекомендуется клонировать шаблон и внести изменения в клон. Генерация Для генерации отчета из шаблона отчета найдите соответствующий шаблон и нажмите кнопку "Сгенерировать". Это позволит оценить шаблон и получить файл в текстовом

формате для скачивания. Сам процесс рендеринга выполняется в фоновом режиме, но может занимать продолжительное время.

Существует отдельное разрешение на генерацию отчета, поэтому только выбранные пользователи могут выполнять ресурсоемкие операции, в то время как другие пользователи могут подготавливать шаблоны отчетов.

Планирование и доставка отчетов по электронной почте В форме генерации отчета вы можете выбрать время начала отложенного рендеринга. Нет способа запланировать повторение, но так как генерацию отчета можно запустить с помощью команды `hammer report-template`, планирование можно осуществить с помощью `cron`. Для запланированных отчетов, а также для долго выполняющихся отчетов, не имеет смысла скачивать отчет веб-браузером. В таких случаях вы можете выбрать доставку отчета по электронной почте, просто установив флажок "Отправить отчет по электронной почте" и указав адрес(а) электронной почты. Ваш адрес электронной почты по умолчанию будет автоматически заполнен, но вы можете указать другие адреса и отправлять отчеты для дальнейшей обработки.

Макросы

Для шаблонов отчетов полезно иметь доступ к большему объему данных из базы данных, чем в обычных шаблонах развертывания. Обычно требуется загрузить все хосты, соответствующие определенному поисковому запросу. Если включен безопасный режим рендеринга, доступ к внутренним объектам ограничен. Центр Управления предоставляет макросы загрузки ресурсов, такие как `load_hosts`. Все они имеют следующие характеристики и возможности:

принимают ключевое слово поиска для ограничения загружаемых ресурсов принимают ключевое слово `include` для указания связанных объектов, которые должны быть предварительно загружены загружают данные пакетами авторизуют ресурсы на основе разрешений текущего пользователя Рассмотрим следующий пример:

```
<%- load_hosts(search: 'name ~ example.com').each_record do |host| -%> <%= host.name %>, <%= host.mac %> <%- end -%>
```

Он загружает все хосты, содержащие домен `example.com` в своем имени. Если таких хостов больше 1000, автоматически будет выполнен отдельный SQL-запрос для каждой тысячи. Затем происходит итерация по всем найденным хостам и выводится пара имя хоста и MAC-адрес, разделенные запятой. Запрос вернет только те хосты, для которых у текущего пользователя есть разрешение `view_hosts`.